IRACST - International Journal of Computer Science and Information Technology & Security (IJCSITS), ISSN: 2249-9555
Vol. 2, No.4, August 2012

Securing the Biometric Image (finger print) through Encryption and Image slicing

P Devaki[#1], Dr. Raghavendra Rao[*2],Kumara swamy[#3]

[#1]*Department of Information Science and Engg,*
The National Institute of Engineering, Mysore, Karnataka, India
[1p]p_devaki1@yahoo.com
[3*2]Department of Computer Science and Engineering,
The National Institute of Engineering, Mysore, Karnataka, India
[2]*grrao57@gmail.com*

***Abstract*— Secret information needs security in a multi user environment. Encryption alone is not sufficient in many cases. In this paper we are providing security to biometric secret image (finger print). The encrypted image is shared using image slicing so that the image need not be stored on the single server instead it is stored on multiple servers and when it is necessary it can be reconstructed and used for the authentication purpose.**

***Keywords*— Decryption, DES, Encryption, image Slicing, secret image, secret sharing.**

I. INTRODUCTION

For some applications it is necessary that the user has to get authenticated himself either by using his password or any biometric features. No doubt that this gives protection against unauthorized entry in to the application or system. But if the password or the biometric feature is stored on a single server then it can not be ruled out that the password or the biometric feature will not be compromised. Because the passwords can be generated by using password generators and the unauthorized user can get in to the system by trying using the generated passwords. In the case of biometric feature like finger print if the attacker is able to get the stored image of the finger and he can get the mould of the finger and he will succeed in entering the system. The key image may be stored on multiple servers as back ups, but it is not far from threats. And also this requires more storage at each place and consistency matters a lot. To avoid this kind of misuse, it is better to store the password or finger print on multiple servers as shares rather than storing the whole image or password on a server. Sharing of secrets provide security because no single share is of any use to the unauthorized user and he may not even know that where all the shares are stored.

Further the security of the secret information can be enhanced by encrypting the password or finger print and then share the password or the finger print and store on distributed servers. This eliminates the single point of failure of the server.

Image slicing is a technique to share the image in to number of shares. Slicing can be done based on bits which are called as bit slicing, and also based on the geometrical axis.

In this work we are encrypting the finger print which is a jpeg image. DES encryption algorithm is being used to encrypt the image. Encrypted image is sliced in to number of shares based on the geometrical axis. Reconstruction requires all the shares. The dealer encrypts the image and divides the encrypted image in to n number of slices. When it is necessary to get in to the system the authorized user needs to provide his finger print image. Once it is given the system can send a request to the other servers to send the slices of the image. Once it collects all the slices it can reconstruct the image. Decrypt the image using the secret key. Compare that reconstructed image with the recently read image for verification. If both the images match with each other then allow the user to get in to the system, other wise take necessary action.

The rest of the paper is organized as follows. Section II explains the DES , section III explains image slicing , section IV proposed work , section V gives the conclusions.

II. DES

DES[11] (Data encryption standard) is one of the symmetric key block encryption algorithms for binary stream of data. The algorithm takes a 56 bits key and 64 bit blocks of plain text to produce blocks of 64 bits cipher text. Only the authorized users will have the symmetric global key to encrypt and decrypt the data.

The structure of DES is such that it is not that easy to break the cipher text with out knowing the key[1][2][3][4].

A. Structure of DES algorithm

DES has 16 rounds; each round has the same structure. Each round does swapping and permutations. This helps in scrambling the bits. The 56 bit key is divided in to 16 sub keys of 48 bits and applied in each round in order. Each sub key is generated by performing permutations and rotations in each round.

Encryption operation:

1. The plain text is the binary data which is given as the input to the algorithm along with 56 bit key. The input is divided in to number of 64 bit blocks.

2. 16 Sub keys are derived from the 56 bit encryption key. Each sub key is of length 48 bits. This is based on bit shuffling. The process is fairly straightforward. For generating each sub key, the previous sub key is halved and the bits of each half are moved one bit to the left. The first bits are wrapped around to the end. The two new halves are rejoined to make a new key. The 56 bit key that you provide to the encryption method is only used to generate the first sub key, and isn't directly used to encrypt the data.

3. Once the plain text has been broken into a series of 64 bit blocks, each block is shuffled (a process also called *permutation*) based on a known 'shuffle' table that specifies how the bit are shuffled. That is, bit 1 is placed in bit 40; bit 2 is placed in bit 23 and so on.

4. The shuffled 64 bits created in step 3 are passed to a round, where it is split into two blocks of 32 bits each and processed against the corresponding key for that round.

5. After the end of each round the 32 bits will be swapped. The swapped bits will be given as input to the next round, and this is continued for all the 16 rounds.

6. Once the 16th round is complete, the resulting two 32 bit halves are swapped and then appended both the blocks into a 64 bit block.

7. Finally, the 64 bit block is then reshuffled (permutated) using the inverse shuffle that was applied in step 3.

All the blocks of the plain text go through this process. Once all the blocks have been processed they are combined, and that's the encrypted cipher text.

Decryption also uses the same algorithm as the DES encryption algorithm except for the order in which the sub keys are applied. In encryption process sub key s1 will be applied in the first round, s2 in the second round and so on up to 16 rounds. Where as in decryption process sub key s16 will be used in the first round, s15 in the second round and so on at the end the s1 will be applied in the 16th round. A simple structure of DES is shown as below.

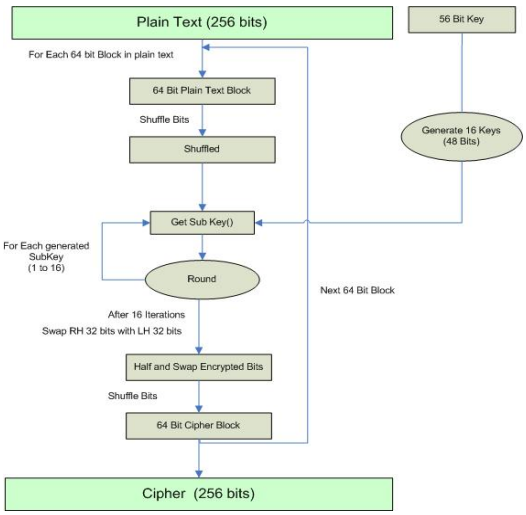

Fig-1 , DES structure

B. Advantage and Disadvantage of DES

The security of DES is quite strong. Even though it was argued that DES was broken in 1990s, but the cost involved was very huge. So still DES can be considered as one of the strongest algorithms, which is because of its structure. It is very easy to implement with hardware. Key length is a concern but additional schemes may be used to enhance the strength of the algorithm.

III. IMAGE SLICING

Image slicing is a technique to divide the image in to number of planes. This is useful in many applications. The planes can be obtained based on bit slicing or geometrical coordinate slicing. Bit slicing is nothing but 8 bit planes are created for all LSBs through MSBs, when each pixel is represented with 8 bits. LSB plane will have all the LSBs from all the pixels of the image; similarly other planes will have the corresponding bits of the image as planes. Combining all these planes will result in the original image. Higher bit planes will have more information about the image than the lower bit planes. So with the few higher bit planes only the image can be reconstructed with out much degradation in quality of the image. Coordinate slicing is to cut the image in X axis or Y axis in to number of planes. Unlike the bit slicing here all the planes are required to reconstruct the image. This is difficult to achieve, because it is necessary to align the slices in the proper position while reconstructing the image.

IV. PROPOSED WORK

Since our aim is to protect the secret key/password/image from unauthorized users, misuse by a single authorized user, we combine the strength of encryption and slicing of image so that only authorized user gets in to the system, with the other servers having the knowledge about the user in a controlled way.

In this work we are concerned about one of the biometric features, finger print to protect from any kind of attack or misuse. If the image is stored on a single server then the attacker may see that the server is down or he may try to damage the image by some means or he may try to get the mould of the image and use it for getting authenticated. So instead of storing that image on a single server, we are proposing a technique where the image can not be copied or damaged and the unauthorized user getting the service. Our work is to first encrypt the image by using DES, then slice that based on the geometrical axis in to number of parts and store each part on different servers. From this even if the attacker succeeds in getting part of the image it's of no use, because first of all it is encrypted and also it is sliced. Hence it does not reveal any information. With only one part or few parts of the image the attacker can not get the mould of the finger and use it for authentication.

In this work the encrypted image is divided in to 4 slices and stored on 4 servers. When a user tries to get authenticated in the system, then the system collects the slices from all the 4 servers and reconstructs the image and matches with the actual

finger impression read by the system for a user. This is referred to as secret sharing[12], because we are sharing the secret image on multiple servers. It is also referred to as multiple keys system. That is each share can be treated as a key. At the time of need it is reconstructed.

Steps for sharing the key image:

1. Encrypt the image
2. Slice the image in to n parts(share)
3. Send each part to different server
4. Each server stores the part in its memory

Steps for reconstruction of the key image:

1. Collect all the shares(parts) from the servers
2. Align the shares to get the encrypted image
3. Decrypt the image by using the symmetric key
4. Use the decrypted image for comparing with the read image

In this application only the authorized user gets the service upon proving his identity.

Same technique can be adopted for passwords or pass phrases also. In the case of passwords/ (pass phrase) any user or system having a single share can request for the other shares from other servers to reconstruct the same.

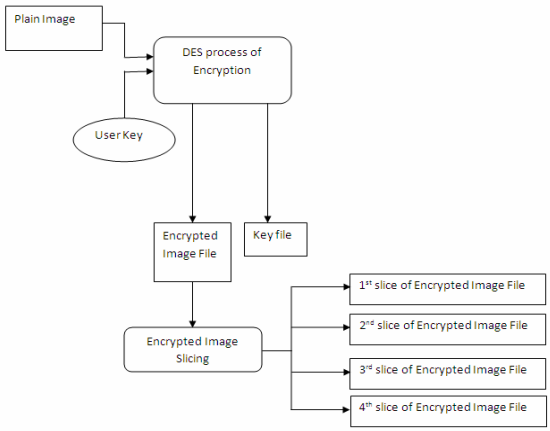

Fig-2, Architecture

A. *Applications*

It can be used in

- Finger print scanners.
- Face recognition, IRIS recognition.
- Secure PCB design, Medical Images [such as cancer].
- Secure the image of secrete agents, bank documents, terrorists and restricted documents of the defense.
- Secure the images of project plan; Aircraft design plan, Ship design plan, etc.

B. *Advantages of our approach:*

- It is highly secured.
- It will not consume more space.
- It will not consume more bandwidth
- The attacker needs both the symmetric key and also all the slices in order to get the finger print, which is very difficult.

C. *Limitations:*

All the slices are required to reconstruct the image, because it is necessary to get the original image with out any degradation in the quality of the image.

D. *Results:*

The image of the finger print is encrypted and sliced in to 4 parts to store on different servers. When the parts are combined and decrypted we obtained the original image. The diagrams related to the experiment and its results are shown in Fig- 3 to Fig – 10.

Fig-3, Original JPG image of 604 × 706 pixels

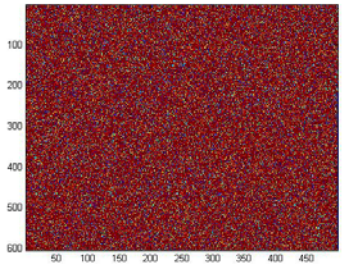

Fig-4, Encrypted image

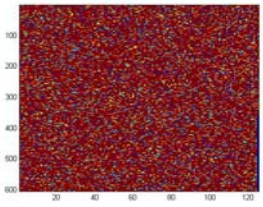

Fig-5, 1st slice

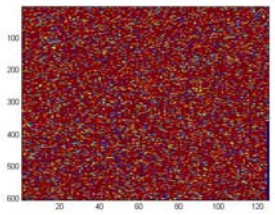

Fig-6, 2nd slice

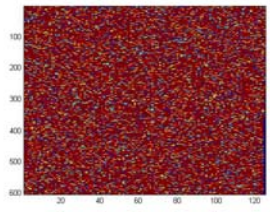

Fig-7, 3rd slice

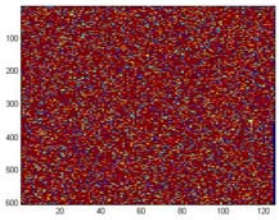

Fig-8,4th slice

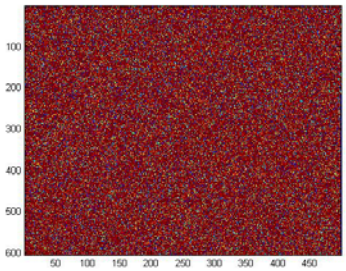

Fig-9, Reconstructed encrypted image

Fig-10, Reconstructed finger print image

V. CONCLUSIONS

With the help of DES and slicing, it is proved that the secrecy of the image can be maintained and also, limitations of storing an image on a single server are eliminated. The result shows that the original image is same as the reconstructed image. This definitely helps the applications in defence, medical, industry and even in other places where security of the image is of more concern. As we can see from the advantages and its uses, this is very important to follow to avoid security attack on a single server. Further the slices can be added with some meaning full image so that the attacker will not get the attention for the encrypted slice. Future work can be using the threshold secret sharing so that with threshold number of slices only it is possible to reconstruct the image.

References:

[1] Biham, Eli and Shamir, Adi, Differential Cryptanalysis of the Data Encryption Standard, Springer Verlag, 1993. ISBN 0-387-97930-1, ISBN 3-540-97930-1.

[2] Biham, Eli and Alex Biryukov: An Improvement of Davies' Attack on DES. J. Cryptology 10(3): 195–206 (1997)

[3] Biham, Eli, Orr Dunkelman, Nathan Keller: Enhancing Differential-Linear Cryptanalysis. ASIACRYPT 2002: pp254–266

[4] Biham, Eli: A Fast New DES Implementation in Software

[5] Campbell, Keith W., Michael J. Wiener: DES is not a Group. CRYPTO 1992: pp512–520

[6] Diffie, Whitfield and Martin Hellman, "Exhaustive Cryptanalysis of the NBS Data Encryption Standard" IEEE Computer 10(6), June 1977, pp74–84

[7] Kaliski, Burton S., Matt Robshaw: Linear Cryptanalysis Using Multiple Approximations. CRYPTO 1994: pp26–39

[8] Knudsen, Lars, John Erik Mathiassen: A Chosen-Plaintext Linear Attack on DES. Fast Software Encryption - FSE 2000: pp262–272

[9] Langford, Susan K., Martin E. Hellman: Differential-Linear Cryptanalysis. CRYPTO 1994: 17–25

[10] S. Tang, "Simple Secret Sharing and Threshold RSA Signature Schemes", *Journal of Information and Computational Science*, 1, 2004, pp. 259–262.

[11] Cryptography and network security by William stallings , 3rd edition, pearson education(page number 37)

[12] G. Blakley, "Safeguarding Cryptographic Keys," *AFIPS Conference Proceedings*, 48, 1979.